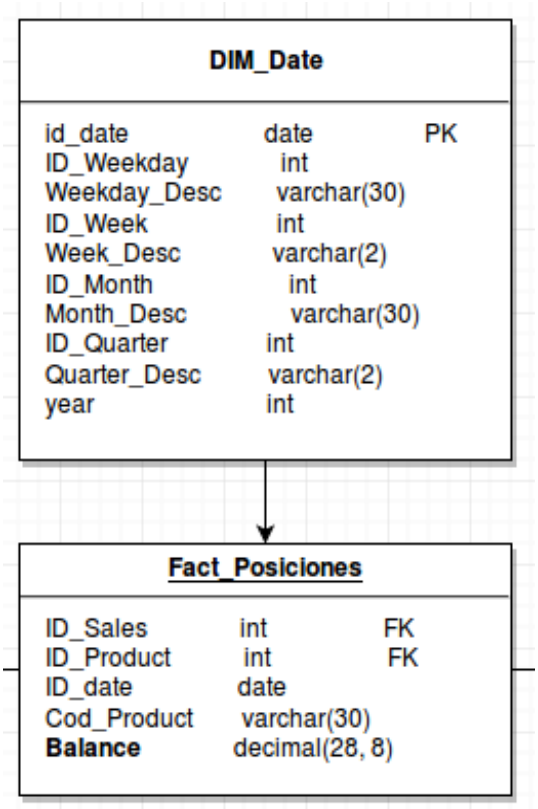
Example of Tuning Cube I

Date: Dicember 2016

Author: Ramón Portolés, Alberto Linkedin a.ramonportoles@gmail.com

Thanks to ShaoFeng Shi for help

Try to optimize a very simple Cube, with 1 Dim and 1 Fact table (Date Dimension)



The baseline is:

- One Measure: Balance, calculate always Max, Min and Count
- All Dim_date (10 items) will be used as dimensions
- Input is a Hive CSV external table
- Output is a Cube in HBase without compression

With this configuration, the results are: 13 min to build a cube of 20 Mb (Cube_01)

Cube_02

To make the first improvement, use Joint and Hierarchy on Dimensions to reduce the cardinality.

Put together all ID and Text of: Month, Week, Weekday and Quarter using Joint Dimension

Define Id_date and Year as a Hierarchy Dimension

This reduces the size down to 0.72 MB and time to 5 min

Kylin 2149, ideally, we can also define these Hierarchies:

- Id_weekday > Id_date
- Id_Month > Id_date
- Id_Quarter > Id_date
- Id_week > Id_date

But for now, it isn't possible to use Joint and hierarchy together in one Dim :(

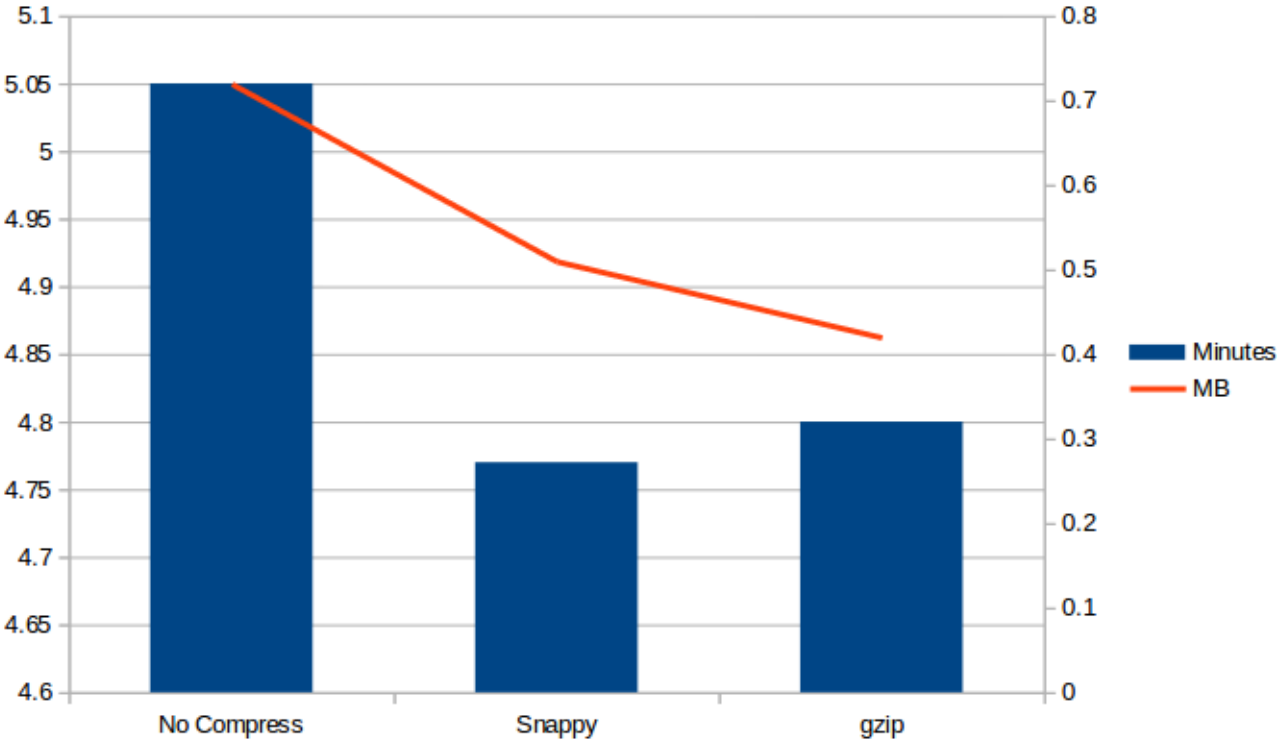
Cube_03

To make the next improvement, compress HBase Cube with Snappy:

Cube_04

Another option is to compress HBase Cube with Gzip:

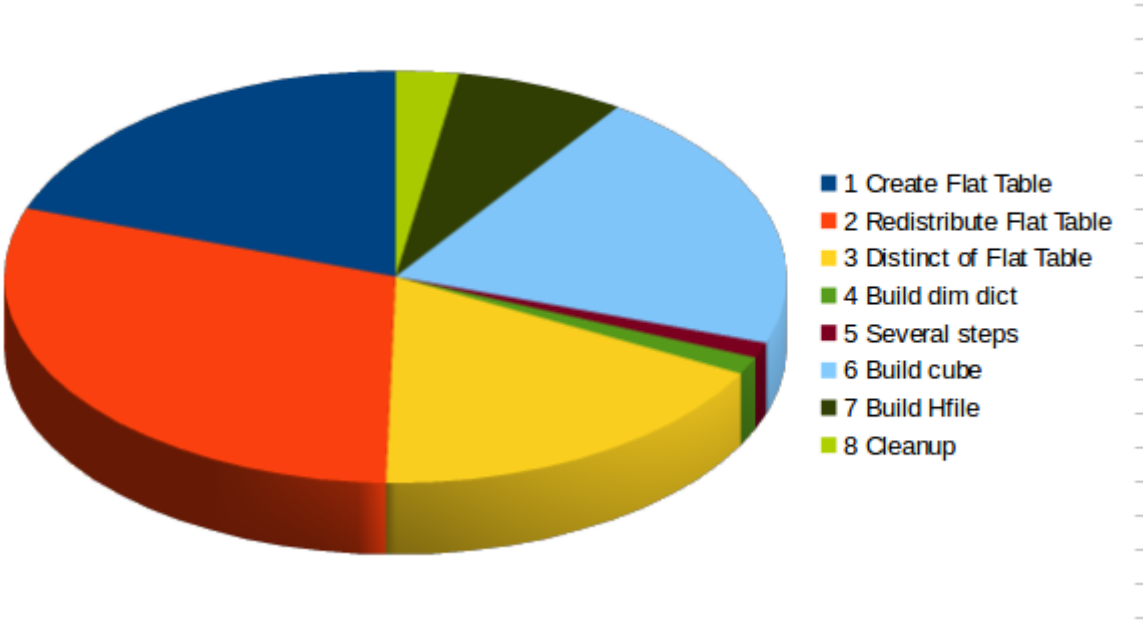
The results of compression output are:



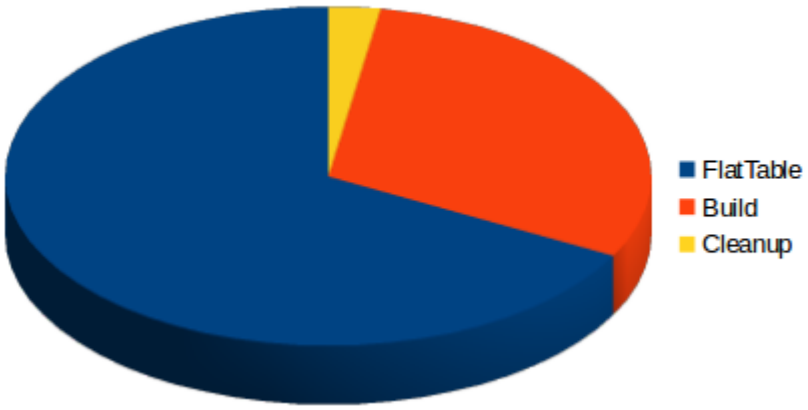
The difference between Snappy and Gzip in time is less than 1% but in size it is 18%

Cube_05

The time distribution is like this:



Group detailed times by concepts :

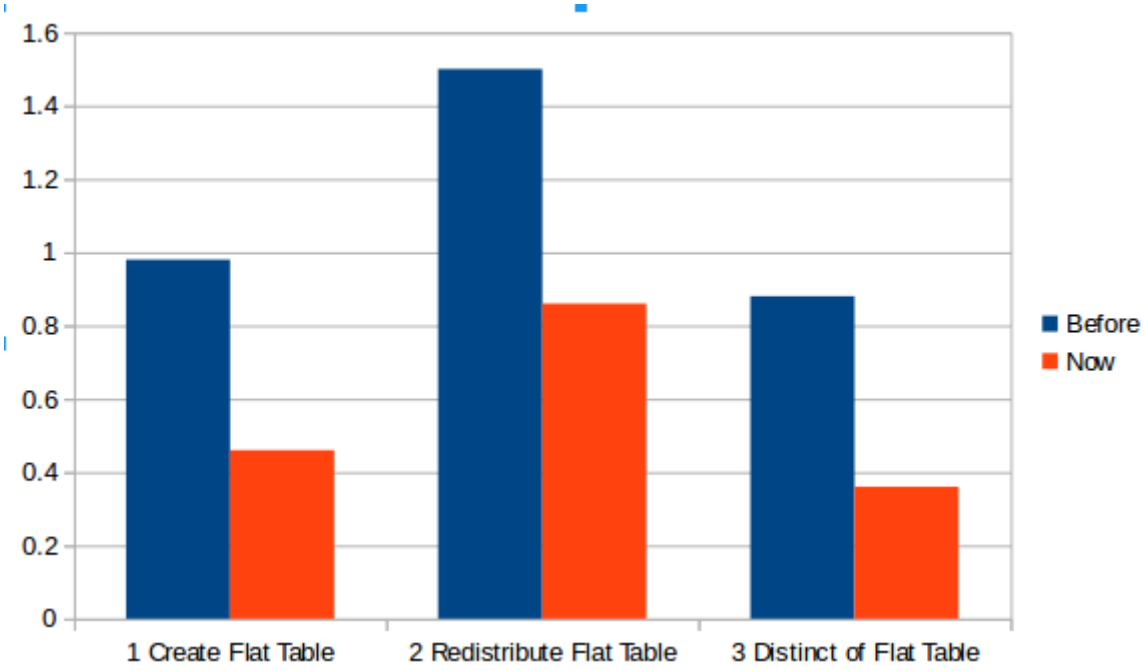


67% is used to build / process flat table and 30% to build the cube

A lot of time is used in the first steps!

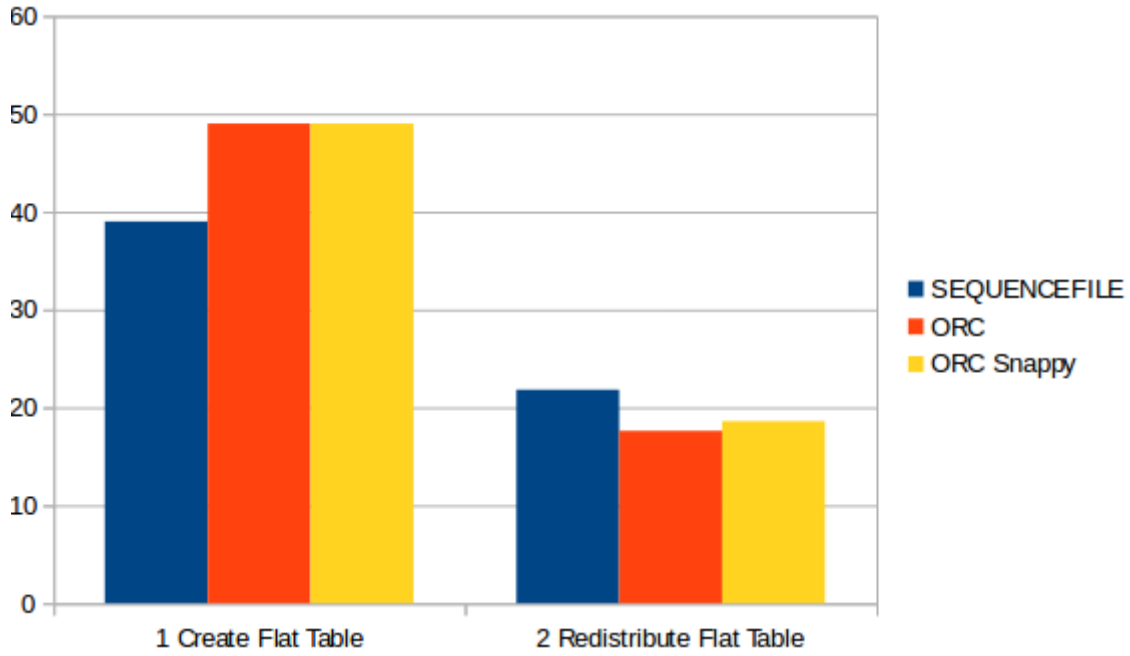
This time distribution is typical in a cube with few measures and few dim (or very optimized)

Try to use ORC Format and compression on Hive input table (Snappy):



The time in the first three steps (Flat Table) has been improved by half :)

Other columnar formats can be tested:

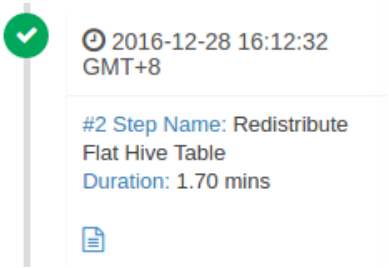


- ORC
- ORC compressed with Snappy

But the results are worse than when using Sequence file …

See comments about this here: Shaofengshi in MailList

The second step is to redistribute Flat Hive table:



Is a simple row count, two approximations can be made

- If it doesn't need to be accurate, the rows of the fact table can be counted→ this can be performed in parallel with Step 1 (and 99% of the time it will be accurate)

```
set hive.exec.compress.output=false;
INSERT OVERWRITE DIRECTORY '/kylin/kylin_metadata/kylin-3b/row_count'
  SELECT count(*) FROM kylin_intermediate_Her_Position_Cube_11_;
```

- See comments about this from Shaofengshi in MailList . In the future versions (Kylin 2265 v2.0), this steps will be implemented using Hive table statistics.

```
set hive.exec.compress.output=false;
INSERT OVERWRITE DIRECTORY '/kylin/kylin_metadata/kylin-3b/row_count'
  SELECT count(*) FROM kylin_intermediate_Her_Position_Cube_11_;
```

Cube_06: Fail

The distribution of rows is:

Fact Table	3.900.00 rows
Dim Date	2.100 rows

And the query (the simplified version) to build the Flat Table is:

SELECT

,DIM_DATE.X

,DIM_DATE.y

,FACT_POSICIONES.BALANCE

FROM FACT_POSICIONES **INNER JOIN** DIM_DATE

ON ID_FECHA = .ID_FECHA

WHERE (ID_DATE >= '2016-12-08' AND ID_DATE < '2016-12-23')

The problem here, is that Hive in only using 1 Map to create Flat Table. It is important to change this behavior. The solution is to partition DIM and FACT in the same columns

- Option 1: Use id_date as a partition column on Hive table. This has a big problem: the Hive metastore is meant for a few hundred of partitions and not thousands (In Hive 9452 there is an idea to solve this, but it isn't finished yet)
- Option 2:Generate a new column for this purpose like Monthslot:

2012-04-26	201204
2012-04-27	201204
2012-04-28	201204
2012-04-29	201204
2012-04-30	201204
2012-05-01	201205
2012-05-02	201205
2012-05-03	201205
2012-05-04	201205

Add the same column to dim and fact tables

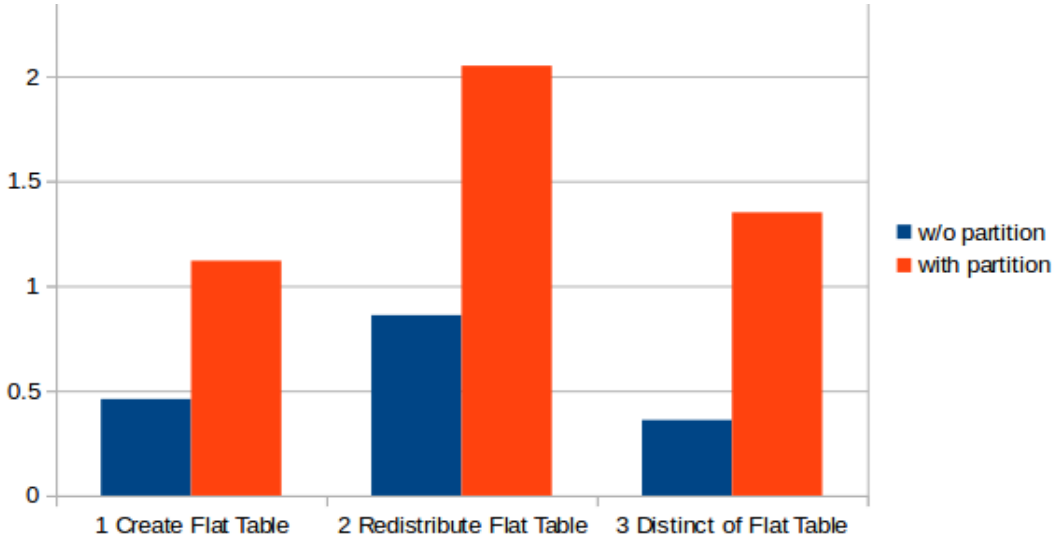
Now, upgrade the data model with this new condition to join tables

The new query to generate flat table will be similar to:

```
SELECT
FROM  FACT_POSICIONES  INNER JOIN DIM_DATE
        ON  ID_FECHA = .ID_FECHA    AND  MONTHSLOT=MONTHSLOT
```

Rebuild the new cube with this data model

As a result, the performance has worsened :(. After several attempts, there hasn't been a solution



The problem is that partitions were not used to generate several Mappers

Task Type	Total	Complete
Map	1	1
Reduce	0	0

(I checked this issue with *ShaoFeng Shi* . He thinks the problem is that there are too few rows and we are not working with a real Hadoop cluster. See this tech note).

Resume of results